# Example: Lotka - Volterra reverse engineering

## MONOMORPHIC RESIDENT POPULATION

**Invasion fitness:**

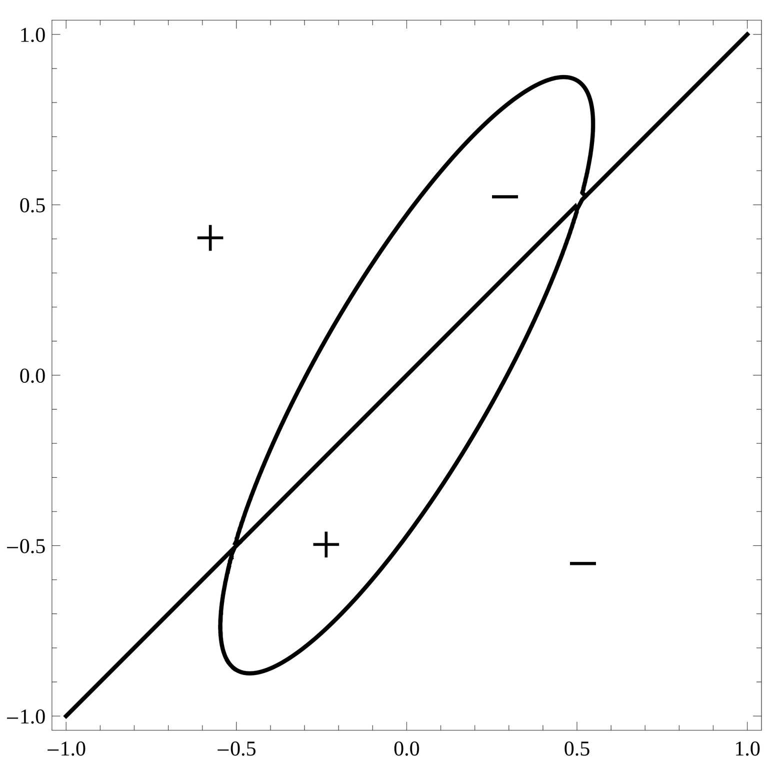
```
s_mo[x_, y_] := (x - y) (1 + α x^2 + β x y + γ y^2);
```

**Default parameter values:**

```
α = -11.5; β = 12.12; γ = -4.5;
```

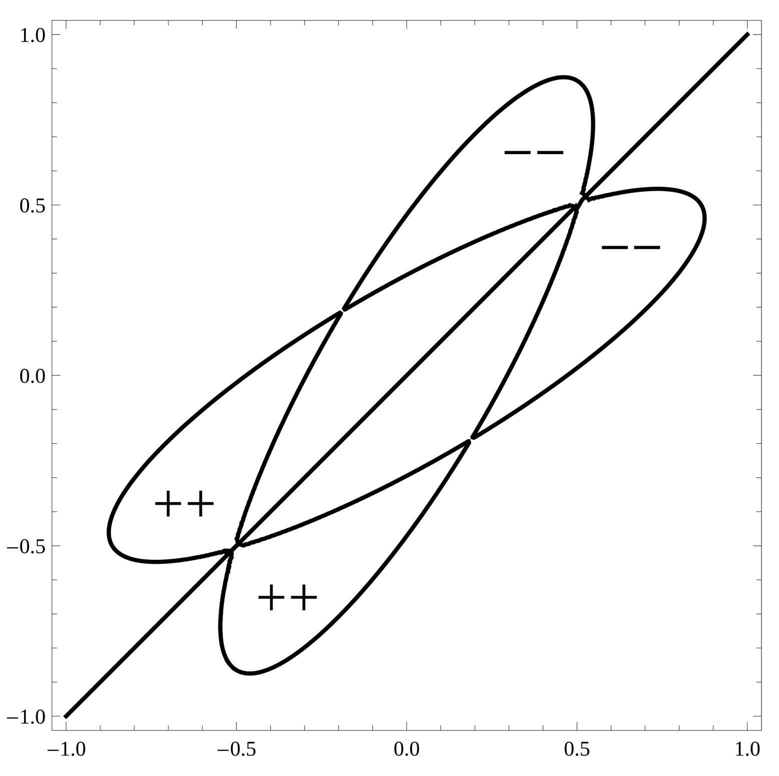
**Pairwise invadability plot (PIP):**

```
ContourPlot[s_mo[x, y], {x, -1, 1}, {y, -1, 1}, Contours → {0}, ContourStyle → {Black, Thick},
 ContourShading → False, PlotPoints → 100]
```




**Mutual invadability plot (MIP) and mutual exclusion plot (MEP):**

```
ContourPlot[s_mo[x, y] s_mo[y, x], {x, -1, 1}, {y, -1, 1}, Contours → {0},
 ContourStyle → {Black, Thick}, ContourShading → False, PlotPoints → 100]
```



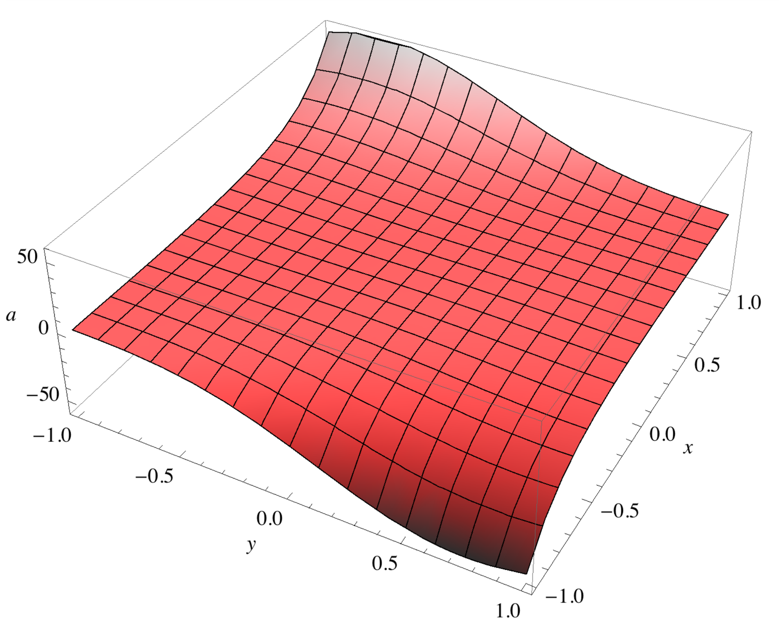

**Default functions for the LV competition model:**

```
r[x_] := 1;
k[x_] := Exp[-x^2];
```

**Corresponding interaction kernel:**

```
a[y_, x_] := (1 - s_mo[x, y] / r[y]) k[y] / k[x];
```

```
Plot3D[a[y, x], {y, -1, 1}, {x, -1, 1}, AxesLabel → {y, x, a}, ColorFunction → "CherryTones",
 PlotRange → {-60, 60}]
```



Possible interpretation: larger trait values expoint smaller trait values.

## DIMORPHIC RESIDENT POPULATION

**Reset:**

```
Clear[α, β, γ, r, k, a];
```

**Dimorphic resident population equilibrium:**

```
Solve[{0 == 1 - (a[x1, x1] n1 + a[x1, x2] n2) / k[x1], 0 == 1 - (a[x2, x1] n1 + a[x2, x2] n2) / k[x2]}, {n1, n2}];
Simplify[%]
```

```
{{n1 → (a[x2, x2] k[x1] - a[x1, x2] k[x2]) / (-a[x1, x2] a[x2, x1] + a[x1, x1] a[x2, x2]), n2 → (a[x2, x1] k[x1] - a[x1, x1] k[x2]) / (a[x1, x2] a[x2, x1] - a[x1, x1] a[x2, x2])}}
```

```
n1[{x1_, x2_}] := (a[x2, x2] k[x1] - a[x1, x2] k[x2]) / (-a[x1, x2] a[x2, x1] + a[x1, x1] a[x2, x2]);

n2[{x1_, x2_}] := (a[x2, x1] k[x1] - a[x1, x1] k[x2]) / (a[x1, x2] a[x2, x1] - a[x1, x1] a[x2, x2]);
```

**Dimorphic invasion fitness and derivatives:**

```
s_di[{x1_, x2_}, y_] := r[y] (1 - (a[y, x1] n1[{x1, x2}] + a[y, x2] n2[{x1, x2}]) / k[y]);
d1s_di[{x1_, x2_}] := (∂_y s_di[{x1, x2}, y]) /. {y → x1};
d2s_di[{x1_, x2_}] := (∂_y s_di[{x1, x2}, y]) /. {y → x2};
dd1s_di[{x1_, x2_}] := (∂_y ∂_y s_di[{x1, x2}, y]) /. {y → x1};
dd2s_di[{x1_, x2_}] := (∂_y ∂_y s_di[{x1, x2}, y]) /. {y → x2};
```

**Default parameter values and functions :**

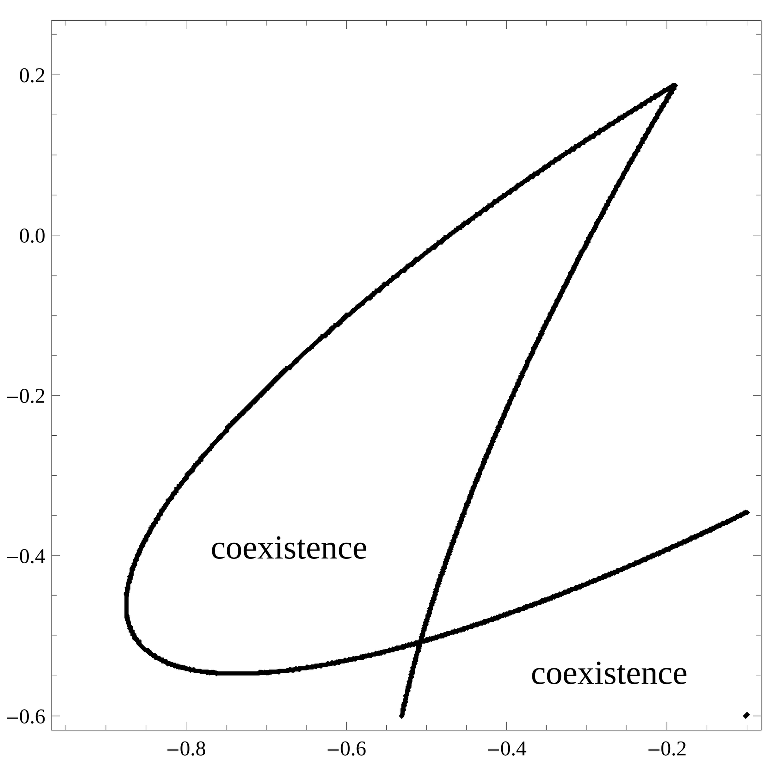
```
r[x_] := 1;
k[x_] := Exp[-x^2];

a[y_, x_] := (1 - s_mo[x, y] / r[y]) k[y] / k[x];

α = -11.5; β = 12.12; γ = -4.5;
```

**Coexistence plot:**

```
coexSet =
 ContourPlot[If[s_mo[x1, x2] > 0 && s_mo[x2, x1] > 0, 1, -1], {x1, -0.95, -0.1}, {x2, -0.6, 0.25},
  Contours → {0}, ContourStyle → {Black, Thick}, ContourShading → False, PlotPoints → 100]
```




**Isocline plot:**

(solid = x1-isocline; dashed = x2-isocline; black = evolutionarily stable; red = not evolutionarily stable)

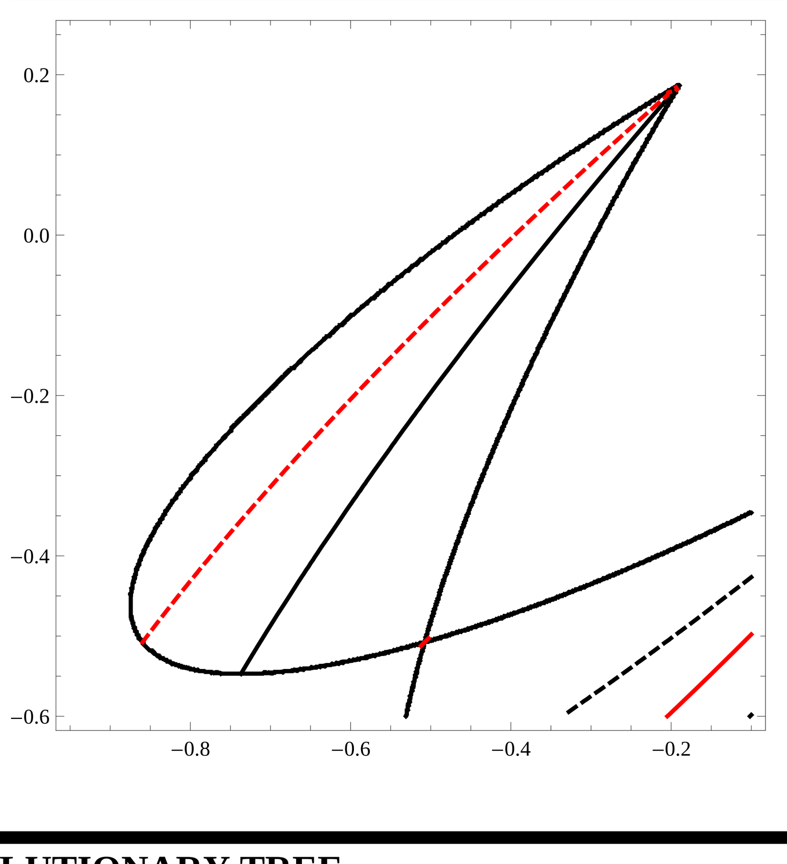
```
iso1ES =
  ContourPlot[If[s_mo[x1, x2] > 0 && s_mo[x2, x1] > 0 && dd1s_di[{x1, x2}] ≤ 0, d1s_di[{x1, x2}]],
   {x1, -0.95, -0.1}, {x2, -0.6, 0.25}, Contours → {0}, ContourShading → False,
   ContourStyle → {Black, Thick}, PlotPoints → 30];

iso1NES =
  ContourPlot[If[s_mo[x1, x2] > 0 && s_mo[x2, x1] > 0 && dd1s_di[{x1, x2}] > 0, d1s_di[{x1, x2}]],
   {x1, -0.95, -0.1}, {x2, -0.6, 0.25}, Contours → {0}, ContourShading → False,
   ContourStyle → {Red, Thick}, PlotPoints → 30];

iso2ES =
  ContourPlot[If[s_mo[x1, x2] > 0 && s_mo[x2, x1] > 0 && dd2s_di[{x1, x2}] ≤ 0, d2s_di[{x1, x2}]],
   {x1, -0.95, -0.1}, {x2, -0.6, 0.25}, Contours → {0}, ContourShading → False,
   ContourStyle → {Black, Thick, Dashed}, PlotPoints → 30];

iso2NES =
  ContourPlot[If[s_mo[x1, x2] > 0 && s_mo[x2, x1] > 0 && dd2s_di[{x1, x2}] > 0, d2s_di[{x1, x2}]],
   {x1, -0.95, -0.1}, {x2, -0.6, 0.25}, Contours → {0}, ContourShading → False,
   ContourStyle → {Red, Thick, Dashed}, PlotPoints → 30];

Show[coexSet, iso1ES, iso1NES, iso2ES, iso2NES]
```



## EVOLUTIONARY TREE

**Individual-based simulated evolutionary tree:**

*(Kisdi et al. Selection 2 (2001) 1– 2,161– 176)*

Note that sometimes the left branch but other times the right branch dies out.